SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Pietro Kristović

Zagreb, 2013.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Troslojna perceptronska mreža učena EBP metodom s adaptivnim koeficijentom učenja

Mentor:
Doc. dr. sc. Danko Brezak

Student:
Pietro Kristović

Zagreb, 2013.

Pod punom moralnom odgovornošću izjavljujem da sam samostalno izradio ovaj rad, koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru, dr. sc. Danku Brezaku na stručnoj pomoći i pružanju korisnih savjeta pri izradi ovog rada.

Zahvaljujem svojoj obitelji na bezuvjetnoj potpori,razumjevanju i strpljenju.

Pietro Kristović

Sadržaj

Sažetak

Metoda povratnog prostiranja pogreške (Error Back-Propagation, EBP) , iako spada među najstarije i dalje je vrlo često korištena metoda učenja višeslojnih perceptonskih mreža. U cilju poboljšanja učenja, u smislu skraćenja vremena učenja, generalizacije i točnosti, razvijene se različite varijante EBP metode,od kojih se u ovom radu razmatra ona sa koeficijentom učenja koji se adaptira tijekom učenja, dok se učenje odvija po skupu. U ovom radu dana je i usporedba karakteristika učenja te metode i klasične varijante EBP metode s fiksnim koeficijentom učenja, kroz niz standardnih testova.

Ključne riječi: Neuronske mreže, EBP metoda, adaptivni koeficijent učenja,učenje po skupu

Popis slika

Popis tablica

Popis oznaka

a	parametar kaotičnog dinamičkog sustava
b	parametar kaotičnog dinamičkog sustava
bias	neuron bez ulaza sa konstantnim izlazom 1
d_k	željena izlazna vrijednost k-tog izlaza neuronske mreže
$\overline{d}$	srednja vrijednost izlazne datoteke učenja
D	izlazni skup datoteka za učenje
$\mathbf{e}$	vektor pogreške
E	suma kvadrata pogreške (funkcija cilja)
I	broj neurona ulaznog sloja mreže
J	broj neurona sakrivenog sloja mreže
K	broj neurona izlaznog sloja mreže
K_p	pojačanje aktivacijske funkcije izlaznog sloja,konstanta pojačanja sustava
N	broj parova ulazno-izlaznih redaka datoteke učenja
$NRMS$	normalizirani korijen srednje kvadratne pogreške (mjera točnosti učenja)
net	vrijednost funkcije sume
net_j	vrijednost funkcije sume j-tog neurona (ulaz j-tog neurona)
net_{Hj}	vrijednost funkcije sume j-tog neurona sakrivenog sloja
net_{Ok}	vrijednost funkcije sume k-tog neurona izlaznog sloja
O_k	k-ti izlaz neuronske mreže
$\mathbf{O}$	vektor (matrica) izlaza iz mreže
T	vremenska konstanta (s)

T_0	period uzorkovanja(s)
$u(n)$	trenutna vrijednost ulaza sustava
v_{ij}	težinski koeficijent veze između j-tog neurona sakrivenog sloja te i-tog neurona ulaznog sloja
$\mathbf{V}$	vektor (matrica) težinskih koeficijenata sakrivenog sloja
w_{kj}	težinski koeficijent veze između k-tog neurona izlaznog sloja i j-tog neurona sakrivenog sloja
η^+	parametar povećanja koeficijenta učenja
η^-	parametar smanjenja koeficijenta učenja
y	vrijednost izlaza neurona
Z_i	i-ti ulaz neuronske mreže
$\mathbf{Z}$	ulazni skup datoteka za učenje
α	vrijednost momentuma prvog reda
ϑ	parametar učenja (težinski faktor)
$\vartheta(n)$	trenutna vrijednost parametra učenja
$\vartheta(n+1)$	nova vrijednost parametra učenja
$\Delta\vartheta(n)$	tekuća promjena parametra učenja
Δ	iznos promjena težina u jednom koraku
δ	parametar algoritma povratnog prostiranja pogreške
δ_{Ok}	parametar algoritma povratnog prostiranja pogreške izlaznog sloja
γ	aktivacijska funkcija neurona
η	koeficijent brzine učenja gradijentnog algoritma
σ_{d_n}	standardno odstupanje (devijacija)
∇E	gradijent pogreške
$\dfrac{\partial E}{\partial w_{ij}}$	parcijalna derivacija gradijenta pogreške po težinskom koeficijentu
τ	vremensko kašnjenje dinamičkog sustava

Poglavlje 1

Uvod

Umjetne neuronske mreže ne predstavljaju novi koncept. Originalna ideja za umjetnu neuronsku mrežu generirala se iz niza pokušaja modeliranja biofiziologije mozga čovjeka, s ciljem razumjevanja i objašnjenja kako isti funkcionira. Ta ideja podrazumijeva kreiranje modela sposobnog da procesira informacije, analogno aktivnostima mozga čovjeka. Najveći dio misaonih procesa mozga čovjeka još uvijek nam je nepoznat, što je realni ograničavajući faktor za razvoj umjetnih neuronskih mreža i umjetne inteligencije općenito. Međutim, postojeće mreže građene na pojednostavljenom modelu (ne uzimaju u obzir većinu poznatih funkcija mozga) pružaju dosta dobre rezultate.

Umjetna neuronska mreža je skup umjetnih neurona koji su međusobno povezani i interaktivni kroz operacije obrade signala. Uređena je po uzoru na ljudski mozak. Za umjetni neuron može se reći da se dizajnira s idejom da oponaša osnovne funkcije biološkog neurona, dok učenje je proces kojim se slobodni parametri neuronske mreže adaptiraju kroz kontinuirani proces stimulacije od okoline u kojoj se mreža nalazi. Vrsta je učenje određena načinom kako se odvija promjena parametara. Ovisno o postavljenom kriteriju mogu se dobiti različite kategorizacije umjetnih neuronskih mreža. Neke od najčešćih podjela neuroskih mreža su: prema protoku signala (unaprijedne i povratne mreže), prema zavisnosti o vremenskim parametrima (statičke i dinamičke mreže), prema načinu učenja (mreže sa učenjem po uzorku i mreže sa učenjem po skupu) te prema učitelju (mreže sa supervizornim i nesupervizornim učenjem). No, postoji i podjela neuronskih mreža na vremenski kontinuirane i vremenski diskretne mreže ili ovisno od implementacije, neuronske mreže mogu biti softverske, hardverske i optičke.

U ovom radu izrađena je programska podrška za učenje različitih troslojnih unaprijednih neuronskih mreža primjenom EBP algoritma s fiksnim i adaptivnim koeficijentom učenja. Pomoću programske podrške su generirani odzivi strukturirane mreže primjenom nekoliko standardnih testova,nakon čega je izvršena njihova usporedba i analiza.

Poglavlje 2

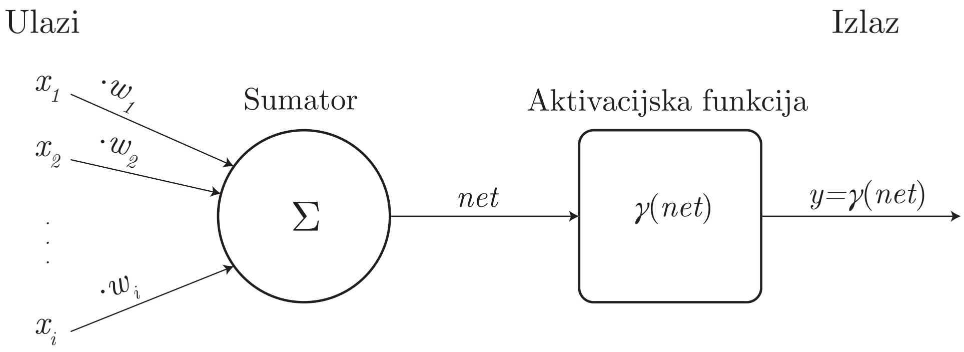
Unaprijedne (statičke) višeslojne neuronske mreže

Unaprijedne višeslojne neuronske mreže definirane su određenom strukturom mreže i smjerom gibanja signala, gdje sloj neuronske mreže predstavlja paralelno složen skup neurona. Uobičajeno je da višeslojne neuronske mreže imaju ulazni i izlazni sloj, dok su između njih tkz. skriveni slojevi. Ako se slojevi neuronske mreže povežu tako da signali putuju samo u jednom smjeru, od ulaza ka izlazima mreže, onda govorimo o unaprijednim(statičkim) neuronskim mrežama(feedforward neural networks).

U ovom radu korištena je spomenuta unaprijedna (statička) višeslojna neuronska mreža. Za potrebe učenja korišten je unaprijed poznat ulazno-izlazni skup učenja i metoda učenja s učiteljem (supervizorno učenje). Navedeni oblik učenja podrazumijeva poznavanje izlaznih i pripadajućih ulaznih vrijednosti mreže, pri čemu algoritam adaptacije parametara mreže ovisi o grešci ili odstupanju dobivenih od željenih izlaznih vrijednosti .

2.1 Model statičkog neurona

Standardni model statičkog neurona prikazan je na slici 2.1. Na slici su prikazane dvije temeljne podfunkcije statičkog neurona, a to su funkcija sume $\sum$ i aktivacijska funkcija γ.



Slika 2.1: *Model statičkog umjetnog neurona*

Kao što je što je prikazano na slici 2.1, vrijednosti ulaza neurona se množe sa pripadajućim težinskim faktorima w i zbrajaju u sumatoru, što je prikazano sljedećom jednadžbom, gdje je $i = 1, 2, 3, ..., n$.

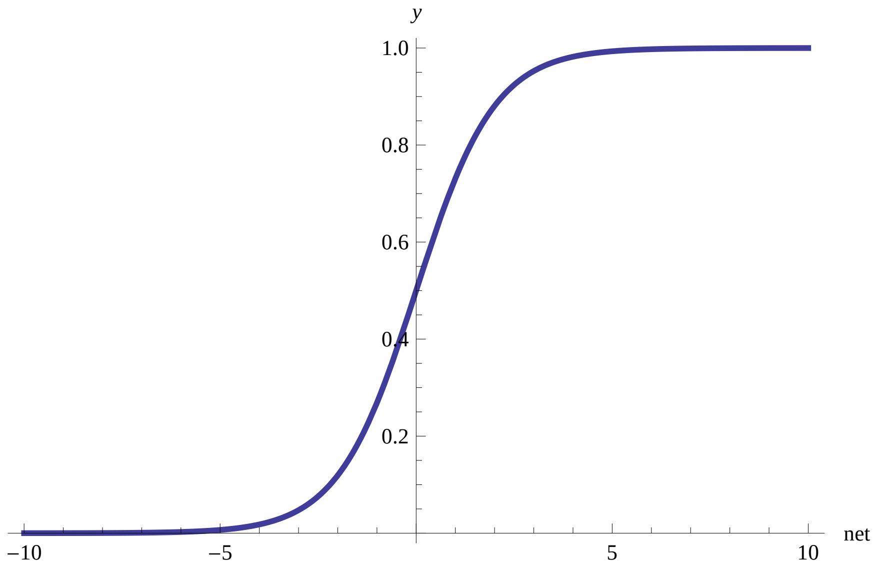
$$net = \sum w_i \cdot x_i \tag{2.1}$$

Aktivacijska funkcija γ vrši preslikavanje rezultata funkcije sume net u izlaznu vrijednost neurona.

$$y = \gamma(net) \tag{2.2}$$

Kad je aktivacijska funkcija oblika praga osjetljivosti, onda se opisani neuron često naziva perceptron. Za aktivacijsku funkciju se najčešće odabire monotono rastuća (derivabilna) funkcija sa zasićenjem. Također, za neurone sakrivenog sloja uvijek se odabire neka nelinearna aktivacijska funkcija, jer u slučaju linearne aktivacijske funkcije ne bi bilo moguće rješavati linearno neseparabilne probleme. Može se reći da učenje neuronske mreže zapravo predstavlja interpolaciju (aproksimaciju) hiperravnine kroz skup točaka za učenje, dok testiranje, tj., korištenje naučene mreže predstavlja računanje vrijednosti aproksimirane hiperravnine za dani ulazni vektor. U ovom radu kao aktivacijsku funkciju ćemo koristiti nelinearnu unipolarnu sigmoidalnu funkciju prikazanu na slici 2.2, koja je jedna od najčešće korištenih aktivacijskih funkcija neurona sakrivenog sloja.
Unipolarana sigmoidalna funkcija definirana je sljedećim izrazom:

$$y(net) = \frac{1}{1 + e^{-net}} \tag{2.3}$$



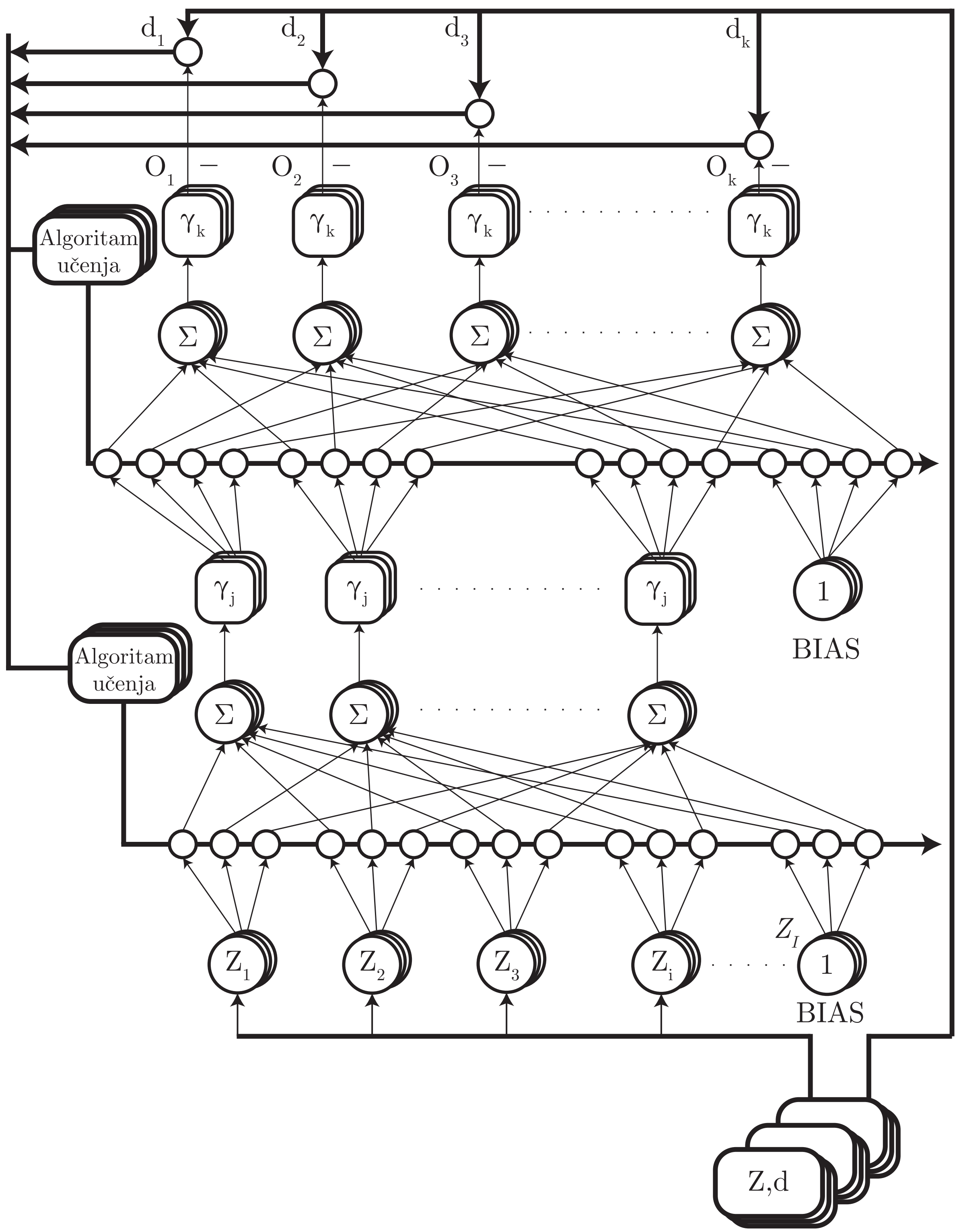
Slika 2.2: *Unipolarna sigmoidalna funkcija*

Primjena odabrane aktivacijske funkcije nužno podrazumijeva i primjenu Bias neurona konstantnog izlaza jednakog jedinici (slika 2.2). Njegovom je primjenom osigurana potrebna pokrivenost područja učenja s aktivacijskim funkcijama neurona skrivenog i izlaznog sloja.

2.2 Model statičke (unaprijedne) neuronske mreže

Struktura neuronske mreže je određena neuronskim slojevima koji su međusobno spojeni vezama opterećenim težinskim koeficijentima. Unaprijedna neuronska mreža je svaka mreža u kojoj su sve veze između slojeva unaprijedne,odnosno, u kojoj su izlazi svakog sloja neurona ulazi isključivo sljedećem sloju neurona.Višeslojna neuronska mreža je svaka ona mreža koja ima barem jedan sakriveni sloj, odnosno, ukupno barem tri sloja – ulazni, sakriveni i izlazni.

Model višeslojne unaprijedne (statičke) neuronske mreže, a ujedno i shematski prikaz postupka učenja iste, prikazuje slika 2.3. Teoremom univerzalne aproksimacije dokazano je kako je jedan sakriveni sloj dovoljan za aproksimaciju bilo koje kontinuirane funkcije, uz uvjet da isti ima dovoljan broj neurona. Međutim, ne postoji *pravilo* po kojem bi se odredio *dovoljan* broj neurona, već samo neke smjernice, koje ovise o problemu. Nepisano je pravilo da broj parametara učenja mora biti dva puta manji od broja uzoraka za učenje, no ovo pravilo ne daje uvijek najbolje rezultate. Veći broj parametara učenja od broja uzoraka za učenje poboljšao bi učenje mreže, ali bi smanjio sposobnost generalizacije. Stoga se broj neurona sakrivenog sloja određuje se eksperimentalno, metodom pokušaja i pogreške.



Slika 2.3: *Model statičke unaprijedne neuronske mreže*

Zadaci na kojima će, u ovom radu, algoritmi učenja biti testirani, jesu regresijski zadaci(identifikacija+predikcija) gdje se od mreže očekuje da, na temelju ulazno - izlaznog skupa podataka, pronađe funkciju koja će dane ulazne podatke preslikati u željene izlazne podatke. Na ulaze mreže djeluju vrijednosti iz datoteka učenja Z, te na temelju toga se izračunavaju vrijednosti izlaza mreže O. Na temelju različitosti dobivenih vrijednosti izlaza mreže i očekivanih vrijednosti d, te koristeći određeni algoritam učenja mijenjaju se parametri mreže, tj. težinski koeficijenti veza w. Neuronske mreže mogu mijenjati parametre na dva različita načina:

1. **Učenje po uzorku** - parametri se mijenjaju na temelju pogreške dobivene prolaskom svakog ulazno-izlaznog para skupa za učenje.

2. **Učenje po skupu** - parametri se mijenjaju nakon što cijeli skup podataka za učenje prođe kroz mrežu i to na temelju izračuna ukupne greške odstupanja stvarnih od željenih izlaznih vrijednosti.

U ovom radu korišteno je učenje po skupu, sa različitim varijantama EBP algoritmima učenja.

2.2.1 Funkcija cilja i ocjena uspješnosti

Razlike izračunatih i željenih izlaza mreže služe nam za izračunavanje vrijednosti funkcije cilja, koja je najčešće definirana kao suma kvadrata tih razlika. Cijeli postupak učenja se ponavlja sve dok se ne postigne pogreška manja ili jednaka dozvoljenoj pogrešci koju ovisno od zadatka zadaje učitelj. To je uobičajena statistička metoda regresijske analize, koja služi kao mjera odstupanja izlaza mreže od željene vrijednosti izlaza. Izraz za sumu kvadrata pogreške glasi:

$$E = \frac{1}{2}\sum_{n=1}^{N}(d_n - O_n)^2 \tag{2.4}$$

gdje je N broj elemenata u skupu za učenje, što je određeno datotekom učenja. Postupak podešavanja težinskih koeficijenata (parametara učenja) podrazumijeva minimiziranje funkcije cilja E određene izrazom (2.4).

Kako bi se procjenila točnost (uspješnost) algoritma učenja u rješavanju postavljenog zadatka, potrebno je definirati tzv. mjeru (indeks) točnosti - bezdimenzionalni normalizirani korijen srednje kvadratne pogreške (tzv. $NRMS$). Izraz za $NRMS$ glasi:

$$NRMS = \frac{\sqrt{\frac{\sum\limits_{n=1}^{N}(d_n - O_n)^2}{N}}}{\sigma_{d_n}}, \tag{2.5}$$

gdje σ_{d_n} predstavlja standardnu devijaciju (odstupanje) definirano kao:

$$\sigma_{d_n} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(d_n - \overline{d})^2}, \tag{2.6}$$

a $\overline{d}$ je:

$$\overline{d} = \frac{1}{N}\sum_{n=1}^{N} d_n. \tag{2.7}$$

Prednost ove mjere točnosti jest njena bezdimenzionalnost koja osigurava neovisnost mjere o dimenzijama učenih veličina, te omogućuje usporedbu izvedenih algoritama učenja s drugim algoritmima.

Poglavlje 3

Učenje povratnim rasprostiranjem greške

Rad umjetne neuronske mreže odvija se u dvije osnovne faze: najprije se odvija faza učenja ili treniranja mreže, a zatim slijedi faza testiranja. Postupak učenja se za svaki korak obavlja u unaprijednoj i povratnoj fazi, te se zasniva se na podešavanju težinskih koeficijenata veza među slojevima i, nerijetko, parametara aktivacijskih funkcija s ciljem da se izlazi mreže, za odgovarajuće ulaze skupa za učenje, što više približe poznatim, željenim vrijednostima izlaza mreže. Od mreže se nikad ne očekuje da izlaz bude identičan željenom, jer kada bi se od mreže tražilo da izlaz bude jednak unaprijed željenom izlazu, tada preslikavanje ulaza na izlaz ne bi više bio problem aproksimacije, nego problem interpolacije. Interpolacija se želi izbjeći jer realni signal često u sebi sadrži šum, što bi značilo da interpolacijom mreža uči šum, interpolirana funkcija prolazi kroz sve točke datoteke učenja, pa se postavlja pitanje kako će se mreža ponašati za točke koje nije naučila(hoće li naučena hiperravnina biti glatka ili će oscilirati).Upitno je i koliko je interpolacija uopće poželjna sa stanovišta računalnih resursa, s obzirom da se u slučaju interpolacije, zahtijeva da suma kvadrata pogreške bude jednaka nuli. Spomenuti razlozi dovode do lošije generalizacije mreže koje su učile interpolaciju od onih koje su učile aproksimaciju funkcije preslikavanja. S obzirom da nema garancije da će pogreška testa, izražena preko $NRMS$-a, padati kako će učenje mreže napredovati, mreža je svakih n koraka mreža je testirana sa trenutnim vrijednostima parametara učenja. Ukoliko se pogreška testa smanjila u odnosu na prethodno izračunatu, parametri učenja su spremljeni, a ukoliko se povećala, parametri nisu uzeti u obzir. U ovom radu bez obzira da li se pogreška testa u danom koraku testiranja smanjila ili povećala, mreža je nastavila dalje sa učenjem, jer je jedini uvjet prekida učenja bio unaprijed zadani broj koraka učenja. Ovakvo testiranje tijekom učenja u literaturi je poznato kao validacija, a skup podataka kojim se mreža testira tijekom učenja kao validacijski skup. Početne vrijednosti težina određujemo generatorom slučajnih brojeva u intervalu u kojemu je normirani skup za učenje. Ako se u početku učenja pojavljuju problemi sa lokalnim minimumom pogreške, onda interval početnih vrijednosti težina treba smanjiti za red veličine.

3.1 Unaprijedna faza učenja

U prvoj, unaprijednoj fazi učenja, iz ulaznog skupa podataka za učenje uzimaju se vrijednosti svih ulaza mreže Z, te se njima izračunava izlaz mreže O. Funkcija sume net neurona skrivenog sloja H, dobiva prvi indeks oznake sloja (net_H), te za svaki j-ti neuron dobiva drugi indeks($net_{Hj} \rightarrow, net_{H1}, net_{H2}, ..., net_{HJ-1}$) i računa se na sljedeći način:

$$net_{Hj} = \sum_{i=1}^{I} v_{ji} \cdot Z_i, \quad j = 1, 2, ...J-1, \quad i = 1, 2, ...I, \tag{3.1}$$

gdje $\boldsymbol{I}$ predstavlja broj ulaznih neurona,a $\boldsymbol{J}$ broj neurona u sakrivenom sloju uvećan za jedan ($bias$). Nakon izračunavanja svih vrijednosti net_{Hj}, vrijednosti izlaza neurona skrivenog sloja su određene aktivacijskom funkcijom, te se izračunavaju prema:

$$y_j = \frac{1}{1 + e^{-net_{Hj}}}, \quad j = 1, 2, ...J-1, \tag{3.2}$$

$$y_J = 1, \quad Bias. \tag{3.3}$$

Izračunate vrijednosti izlaza neurona sakrivenog sloja preko težinskih koeficijenata w_{kj} spojene su na ulaz svakog neurona izlaznog sloja. Sada funkcija sume net neurona izlaznog sloja $\boldsymbol{O}$ dobiva prvi indeks pripadnog sloja (net_O), te drugi pripadni indeks svakom neuronu izlaznog sloja($net_{Ok} \rightarrow, net_{O1}, net_{O2}, ..., net_{OK}$) ,a može se prikazati izrazom:

$$net_{Ok} = \sum_{j=1}^{J} w_{kj} \cdot y_j, \quad k = 1, 2, ...K, \tag{3.4}$$

gdje K predstavlja broj neurona izlaznog sloja, odnosno broj izlaza mreže. Za aktivacijsku funkciju izlaznog sloja odabrana je jedinična linearna funkcija, pa je izlaz neurona sakrivenog sloja definiran kao:

$$O_k = K_p \cdot net_{Ok} = net_{Ok}, \quad K_p = 1. \tag{3.5}$$

gdje je K_p nagib linearne aktivacijske funkcije. Ovim izračunavanjem vrijednosti izlaza mreže završava unaprijedna faza učenja.

3.2 Povratna faza učenja

U drugoj fazi, na osnovu razlike stvarnog i željenog izlaza mreže računa se pogreška učenja na način kako je prikazano u izrazu (2.4). Na osnovu pogreške učenja vrši se korekcija vrijednosti težinskih koeficijenata veza među slojevima. Poznata forma promjene parametara ϑ(težinskih koeficijenata) sljedećeg je oblika:

$$\vartheta(n+1) = \vartheta(n) + \Delta\vartheta(n), \tag{3.6}$$

gdje je n trenutni korak učenja, $\Delta\vartheta(n)$ veličina promjene parametra učenja. Kao što je već rečeno, korišten je EBP (eng. *Error Back Propagation*) algoritam učenja neuronskih mreža, koji predstavlja najpoznatiji i najčešće primjenjivani način promjene parametara učenja. Pogrešku $E(\vartheta)$ moguće je u okolišu točke ϑ aproksimirati s prva dva člana Taylorovog reda:

$$E(\vartheta + \Delta\vartheta) \approx E(\vartheta) + \Delta E(\vartheta) \tag{3.7}$$

$$\Delta E(\vartheta) = \Delta\vartheta^T \nabla E(\vartheta) \tag{3.8}$$

Izraz (3.8) naziva se gradijentom pogreške. Da bi se greška smanjivala najvećim mogućim iznosom, treba odrediti $\Delta\vartheta$ za koji promjena pogreške učenja $\Delta E(\vartheta)$ poprima najveći mogući iznos, a to se ostvaruje uz uvjet:

$$\Delta\vartheta = -\eta \nabla E(\vartheta) \tag{3.9}$$

gdje je η mjera te promjene, koja se još naziva koeficijentom učenja. Stoga izraz (3.6) se može napisati u obliku:

$$\vartheta(n+1) = \vartheta(n) - \eta\nabla E(\vartheta). \tag{3.10}$$

Iako je promjena parametara učenja vrlo jednostavna, ona ima nekoliko nedostataka. Naime, ako se bolje promotri pravilo (3.10), vidi se kako je utjecaj gradijenta pogreške na promjenu parametara učenja zapravo suprotan intuiciji. Ako je gradijent funkcije pogreške mali, parametri učenja će se sporo mijenjati, a poželjno je upravo suprotno – što veća i brža promjena parametara učenja, kako bi se što prije izašlo iz područja . S druge strane, ako je funkcija pogreške strma, tj. gradijent pogreške velik, parametari učenja će se mijenjati brzo, odnosno, sa velikim skokovima, a u ovom je slučaju poželjna mala promjena parametara kako mreža ne bi preskočila minimum funkcije pogreške ili ušla u oscilacije oko istog. Naposljetku, za sve parametre učenja koristi se isti koeficijent brzine učenja, što dovodi do poteškoća u odabiru istog, jer je teško pronaći jedinstveni koeficijent brzine učenja koji bi podjednako dobro utjecao na sve parametre učenja. Odabir ovog koeficijenta proizvoljan je te ovisi kako o problemu, tako i o učitelju, a u literaturi se preporuča uzeti ovaj koeficijent iz intervala $[0.001, 10]$. Zbog problema u učenju uzrokovanim fiksnim koeficijentom učenja, uvodi se adaptivni koeficijent učenja . Postoje više načina promjene koeficijenta učenja, poput Delta-Delta,Delta-Bar-Delta i SuperSAB. U ovom radu biti će korišten modificirani SuperSAB, čije je pravilo promjene koeficijenta učenja u koraku k dano izrazom:

$$\eta_{ij}(k) = \begin{cases} \eta^{+} \cdot \eta_{ij}(k-1), & \text{ako je } \dfrac{\partial E}{\partial w_{ij}}(k-1) \cdot \dfrac{\partial E}{\partial w_{ij}}(k) > 0 \\ \eta^{-} \cdot \eta_{ij}(k-1), & \text{ako je } \dfrac{\partial E}{\partial w_{ij}}(k-1) \cdot \dfrac{\partial E}{\partial w_{ij}}(k) < 0 \\ \eta_{ij}(k-1), & \text{ako je } \dfrac{\partial E}{\partial w_{ij}}(k-1) \cdot \dfrac{\partial E}{\partial w_{ij}}(k) = 0 \end{cases} \tag{3.11}$$

pri čemu vrijedi $0 < \eta^{-} < \eta^{+}$(predložene vrijednosti ovih koeficijenata su 0.5 i 1.05),dok je η_{ij} koeficijent učenja za težinski koeficijent w_{ij},čija vrijednost ima ograničenje $\eta_{max} = 10$. Poznato je da je najveći nedostatak algoritma povratnog rasprostiranja pogreške veliki broj potrebnih iteracija (koraka učenja). Osim toga, ovisnost promjene parametara učenja direktno o gradijentu pogreške može biti vrlo problematično jer ponašanje gradijenta može biti vrlo promjenjivo, što dovodi do poteškoća u nalaženju minimuma pogreške - može se pojaviti zaglavljivanje u minimumu funkcije. Da bi se umanjili ovi problemi, a i ubrzao proces učenja, što znači da se za istu dozvoljenu pogrešku učenja smanji potreban broj koraka učenja, pribjegava se modifikacijama algoritma. U literaturi se često predlaže modifikacija algoritma promjene parametra učenja danog jednadžbom (3.9) na način da se uvede tzv. *momentum* (količina gibanja), odnosno zamah. Modificirana jednadžba (3.9) sada glasi:

$$\Delta\vartheta(n) = -\eta \nabla E(\vartheta(n)) + \alpha \Delta\vartheta(n-1), \tag{3.12}$$

gdje n označava trenutnu promjenu parametara učenja, a $(n-1)$ prethodnu promjenu parametra učenja. Vrijednost koeficijenta momentuma α proizvoljna je i određuje ju učitelj, a obično se bira u intervalu između 0.1 i 0.9. Naime, vrijednost momentuma α manja od 0.1 nema značajnog utjecaja, dok vrijednost veća od 0.9 dovodi do problematičnog ponašanja, pojavljuju se oscilacije uslijed prevelikog utjecaja prethodne promjene parametra učenja. Uvođenje momentuma može ubrzati klasični algoritam rasprostiranja pogreške i do 10 puta. Promjena parametra učenja (3.10) primjenom momentuma poprima konačan oblik:

$$\vartheta(n+1) = \vartheta(n) - \eta \nabla E(\vartheta(n)) + \alpha \Delta\vartheta(n-1). \tag{3.13}$$

EBP algoritam s adaptivnim koeficijentom učenja (uključujući unaprijednu fazu učenja):

(1) Inicijalizacija parametara učenja i odabir vrijednosti koeficijenata učenja (η_{ij}, η^{-},η^{+}, α)

(2) Definiranje kriterija prekida učenja - maksimalni broj koraka učenja, željena točnost ili neki drugi kriterij

(3) Resetiranje brojača koraka učenja - postavljanje na 0

(4) Započimanje petlje koja je aktivna sve dok se ne zadovolji kriterij prekida učenja, a funkcionira na slijedećem principu:

  (a) Započinje novi korak

  (b) Postavlja brojač ulazno-izlaznih uzoraka za učenje na $n = 1$

  (c) Provlači ulaz n-tog ulazno-izlaznog uzorka za učenje kroz mrežu (unaprijedna faza) , te izračunava izlaz mreže

  (d) Uspoređuje željeni i dobiveni izlaz iz mreže i na temelju toga određuje pogrešku

(e) Računa gradijent pogreške po parametrima učenja, i to na sljedeći način:

- Težine izlaznog sloja:

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial O_k}\frac{\partial O_k}{\partial net_{Ok}}\frac{\partial net_{Ok}}{\partial w_{kj}} = -(d_k - O_k)\cdot y_j \tag{3.14}$$

- Težine sakrivenog sloja:

$$\begin{aligned}\frac{\partial E}{\partial v_{ij}} &= \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial net_{Hj}}\frac{\partial net_{Hj}}{\partial v_{ij}} \\ &= y_j\,(1 - y_j)\left(-\sum_{k=1}^{K}(d_k - O_k)w_{kj}\right)\cdot Z_i\end{aligned} \tag{3.15}$$

(f) Sprema gradijente pogreške svakog težinskog koeficijenta

(g) Povećava brojač ulazno-izlaznih uzoraka za učenje za 1, $n = n + 1$

(h) Vraća se na 4c, osim u slučaju kada su svi ulazno-izlazni uzorci primjenjeni na mreži ($n = N$). U tom slučaju ide na 4i.

(i) Računaju se konačni gradijenti pogreške za svaki težinski koeficijent,spremaju se, te na temelju izračunatih gradijenata pogreške i gradijenata pogreške prošlog koraka računaju se koeficijenti učenja prema (3.11).

(j) Parametri učenja se mijenjaju prema izrazu (3.13)

(k) Provjerava točnost naučenog pomoću izraza (2.5)

(l) Provjerava da li je zadovoljen koji od uvjeta prekida učenja - ako jest, učenje se završava, ako nije, tada povećava brojač koraka za 1.

(m) Vraća se na 4a i ponavlja 4a do 4l toliko dugo dok neki od kriterija prekida učenja ne bude zadovoljen.

Poglavlje 4

Opis testova neuronske mreže

Analiza neuronske mreže i dviju varijanti EBP metode učenja provedena je primjenom niza standardnih testova, učenjem ponašanja jednostavnog linearnog dinamičkog P1 člana i nelinearnog kaotičnog sustava poznatog pod nazivom Glass-Mackeyeva jednadžba.

4.1 Linearni dinamički sustav(P1 član)

Dinamika P1 člana glasi:

$$T\frac{\partial x(t)}{\partial t} + x(t) = K_p u(t) \tag{4.1}$$

gdje je T vremenska konstanta, a K_p konstanta pojačanja sustava. Za učenje je potrebno jednadžbu (4.1) diskretizirati:

$$x(n) = \frac{T_0}{T + T_0}\left[\frac{T}{T_0}x(n-1) + K_p u(n)\right] \tag{4.2}$$

gdje $x(n)$ i $u(n)$ označavaju trenutne vrijednosti izlazne i ulazne veličine u n-tom koraku, $x(n-1)$ označava vrijednosti izlazne veličine u $(n-1)$ koraku, pri čemu su koraci određeni periodom uzorkovanja T_0. Uz zadan početni uvjet $x(0)$ i vektor ulaza $u(n)$, ponašanje sustava je određeno, samim time i ulazno-izlazni skup za učenje. Neuronska mreža će imati dva ulaza, na jednom će biti vrijednosti $u(n)$, a na drugom $x(n-1)$, dok je željena vrijednost izlaza $x(n)$, čime je ova neuronska mreža prividno dinamička(povratna). Prema [1] odabrane su vrijednosti parametara:

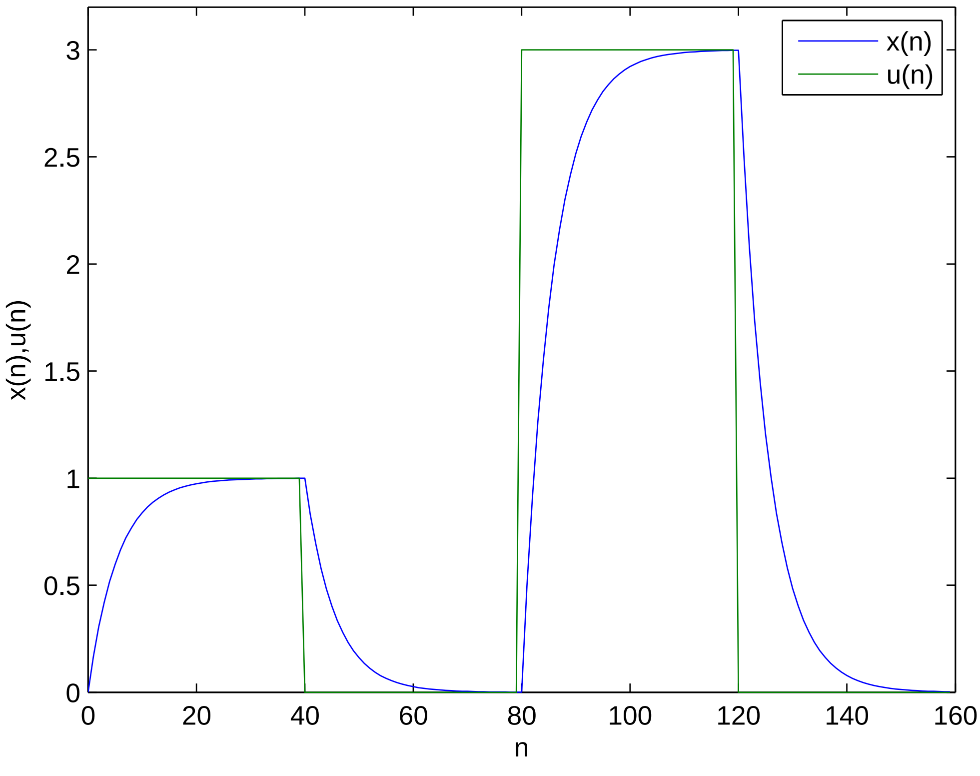
- $T = 1 \quad \text{s}$
- $T_0 = 0.2 \quad \text{s}$

- $K_p = 1$

Datoteku učenja definira odabrani ulazni vektor $u(n)$:

$$u(n) = \begin{cases} 1 & , n = 0, 1, 2, ..., 39 \\ 0 & , n = 39, 40, ..., 79 \\ 3 & , n = 80, 81, ..., 119 \\ 0 & , n = 120, 121, ..., 159 \end{cases} \tag{4.3}$$

Odziv linearnog(P1 člana) na pobudu definiranu izrazom (4.3), ujedno i grafički prikaz ulaza mreže dan je na slici 4.1.



Slika 4.1: *Prikaz skupa uzoraka za učenje P1 dinamičkog člana*

Nakon što mreža završi sa učenjem, testirati će se sa 5 testova, dok je svaki od njih definiran na sljedeći način:

1.

$$u(n) = \begin{cases} 0.5 & , n = 0, 1, 2..., 39 \\ 0 & , n = 40, 41, ..., 79 \end{cases} \tag{4.4}$$

2.

$$u(n) = \begin{cases} 1 & , n = 0, 1, 2..., 39 \\ 0 & , n = 40, 41, ..., 79 \end{cases} \tag{4.5}$$

3.

$$u(n) = \begin{cases} 2 & , n = 0, 1, 2..., 39 \\ 0 & , n = 40, 41, ..., 79 \end{cases} \tag{4.6}$$

4.

$$u(n) = \begin{cases} 2.5 & , n = 0, 1, 2..., 39 \\ 0 & , n = 40, 41, ..., 79 \end{cases} \tag{4.7}$$

5.

$$u(n) = \begin{cases} 3 & , n = 0, 1, 2..., 39 \\ 0 & , n = 40, 41, ..., 79 \end{cases} \tag{4.8}$$

Za validaciju se koristio ulazno-izlazni skup, određen dinamikom i ulaznom funkcijom P1 člana, koja je dana izrazom ispod.

$$u(n) = \begin{cases} 1.5 & , n = 0, 1, 2....39 \\ 0 & , n = 40, 41, ...79 \end{cases} \tag{4.9}$$

4.2 Nelinearni kaotični sustav(Glass-Mackey)

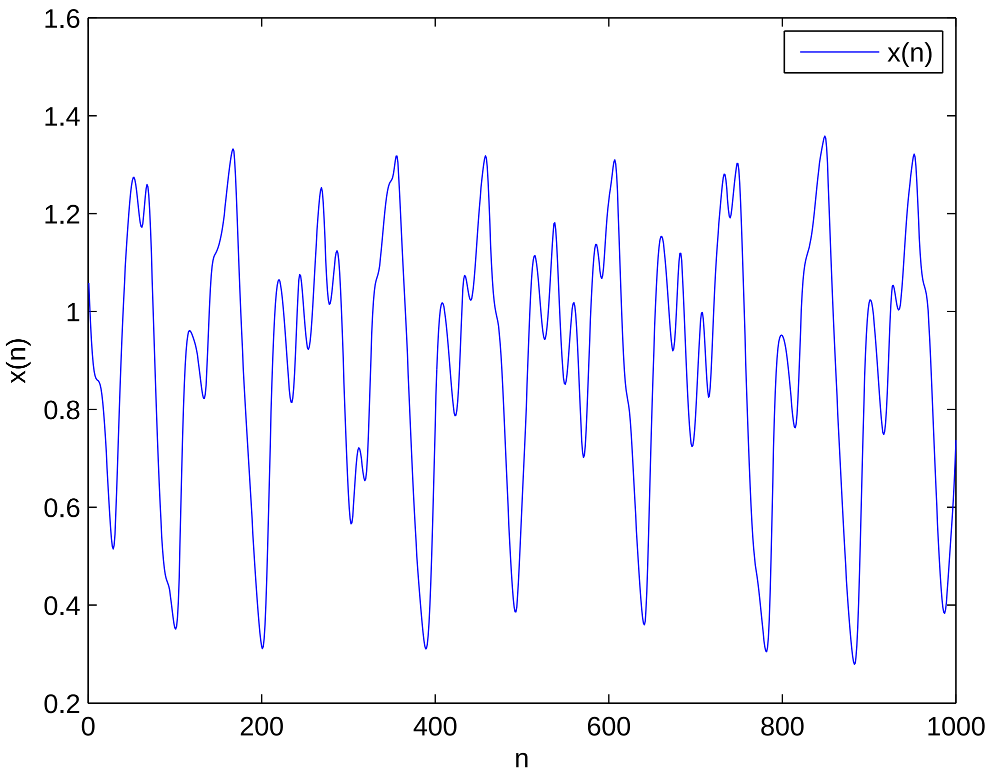
Glass-Mackeyeva jednadžba dana izrazom (4.10) nelinearna je diferencijalna jednadžba sa kašenjenjem za interval τ, koja je određena početnom funkcijom definiranom kako slijedi:

$$\frac{\partial x(t)}{\partial t} = \frac{a \cdot x(t-\tau)}{1 + x^{10}(t-\tau)} - bx(t) \tag{4.10}$$

gdje su a i b parametri sustava. U diskretnom obliku,uzmemo li period uzorkovanja signala $T_0 = 1s$, izraz (4.10) glasi:

$$x(n) = \frac{1}{1+b}\left[x(n-1) + \frac{a \cdot x(n-\tau)}{1 + x^{10}(n-\tau)}\right] \tag{4.11}$$

Odaberemo li vrijednosti parametara $a = 0.2$,$b = 0.1$, $\tau = 30$, te $x(t) = 1.2$ za $t \leq 0$, funkcija $x(n)$ se za 1000 točaka(bez prvih 100 točaka zbog izbjegavanja utjecaja



Slika 4.2: *Prikaz Glass-Mackey vremenske serije, uz* $\tau = 30$

početnih uvjeta) se grafički može prikazati slikom 4.2. Osnovni zadatak predviđanja ponašanja nelinearnog kaotičnog sustava svodi se na uporabu prošlih vrijednosti signala $x(n)$(do neke vremenske točke) za predikciju buduće vrijednosti $x(n+P)$(za P točaka u budućnosti). Standardna metoda za postizanje ovakvog tipa predikcije je u određivanju preslikavanja $f(\bullet)$ kako slijedi:

$$x(n + P) = f(x(n), x(n - \Delta), (x - 2\Delta), ..., x(n - m\Delta)) \tag{4.12}$$

gdje je P broj točaka vremena za predviđanje u budućnosti,Δ je kašnjenje signala, a m cjelobrojna konstanta. Odabirom $P = \Delta$ moguće je predvidjeti ponašanja sustava P koraka u budućnost na način da se vrijednosti izlaza sustava određenim ritmom vraćaju ponovno na ulaz. Za $P = \Delta = 6$ i $m = 4$ izraz (4.12) dobiva slijedeću formu:

$$x(n + P) = f(x(n), x(n - \Delta), (x - 2\Delta), ..., x(n - m\Delta)). \tag{4.13}$$

Funkcija $f(\bullet)$ se u ovom radu aproksimira zadanom neuronskom mrežom, zbog čega mreža ima $m+1$ ulaza(bez bias-a). Učenje i testiranje mreže ostvareno je podjelom vremenske serije (slika 4.2) na dva jednaka dijela - prvih 500 točaka korišteno je se za učenje,ostalih 500 za validaciju, a potom je rad naučene mreže testiran na tri niza od 500 točaka koji su se razlikovali po početnim stanjima.

Poglavlje 5

Rezultati

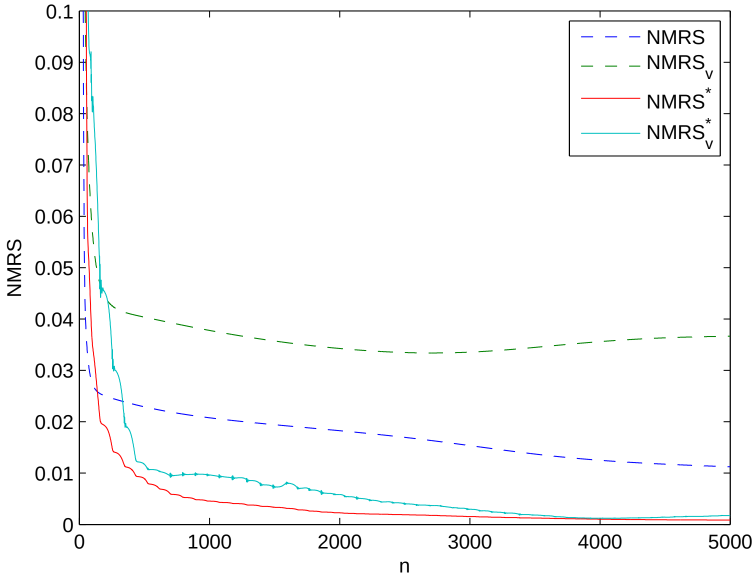
U prošlom poglavlju su opisani testovi pomoću kojih ćemo usporediti rad troslojene perceptronske neuronske mreže učene EBP algoritom sa adaptivnim koeficijentom učenja i EBP algoritom sa fiksnim koeficijentom učenja. Pri učenju ponašanja P1 člana u skrivenom sloju su postavljena tri neurona(bez bias-a), dok je pri učenju predviđanja ponašanja nelinearnog kaotičnog sustava opisanog Glass-Mackeyevom jednadžbom korišteno pet neurona(bez bias-a) . Broj ulaza i izlaza, odnosno, broj ulaznih i izlaznih neurona ovisi o problemu.Brzina i kvaliteta učenja međuostalim ovisi o redu veličine ulaznih i izlaznih vektora, o njegovom odnosu sa početnim težinama, te odabranoj aktivacijskoj funkciji. Normiramo li ulazne vrijednosti u granicama $[0, 1]$,te koristimo unipolarnu sigmoidalnu aktivacijsku funkciju čiji je raspon vrijednosti $[0, 1]$, početne težine odabiremo nasumično u granicama $[0, 0.1]$. Uz učenje se primjenjivala validacija koja se izvršavala svakih 100 koraka učenja. Učenje ponašanja P1 člana je izvršeno u 5000 koraka, a učenje ponašanja nelinearnog kaotičnog sustava u 10000 koraka. Učenje mreže sa različitim varijantama EBP algoritma i njeno testiranje, provođeno je u programskom paketu *MATLAB*. Prilikom učenja i testiranja mreža korišteni su sljedeći parametri algoritama:

1. **EBP sa adaptivnim koeficijentom učenja, učenje po skupu**
   - $\eta_{ij} = 0.004;\ \eta^{+} = 1.05;\ \eta^{-} = 0.5;\ \alpha = 0.7$
2. **EBP, učenje po skupu**
   - $\eta = 0.002;\ \alpha = 0.8$

Tablice 5.1 i 5.2, te slike 5.1 i 5.2 prikazuje rezultate dobivene učenjem, validacijom i testiranjem mreže na zadanim problemima.

Tablica 5.1: *Rezultati učenja mreže na problemu identifikacije linearnog dinamičkog člana(EBP^* - EBP sa adaptivnim koeficijentom učenja;NMRS - NMRS učenja;$NMRS_v$ - NMRS validacije; $NMRS_{ti}$ - NMRS i-tog testa).*

	NMRS	$\mathbf{NMRS_v}$	korak	$\mathbf{NMRS_{t1}}$	$\mathbf{NMRS_{t2}}$	$\mathbf{NMRS_{t3}}$	$\mathbf{NMRS_{t4}}$	$\mathbf{NMRS_{t5}}$
EBP*	0.0008	0.0017	5000	0.0047	0.0016	0.0014	0.0008	0.0008
EBP	0.0096	0.0367	5000	0.0662	0.0169	0.0486	0.0359	0.0121

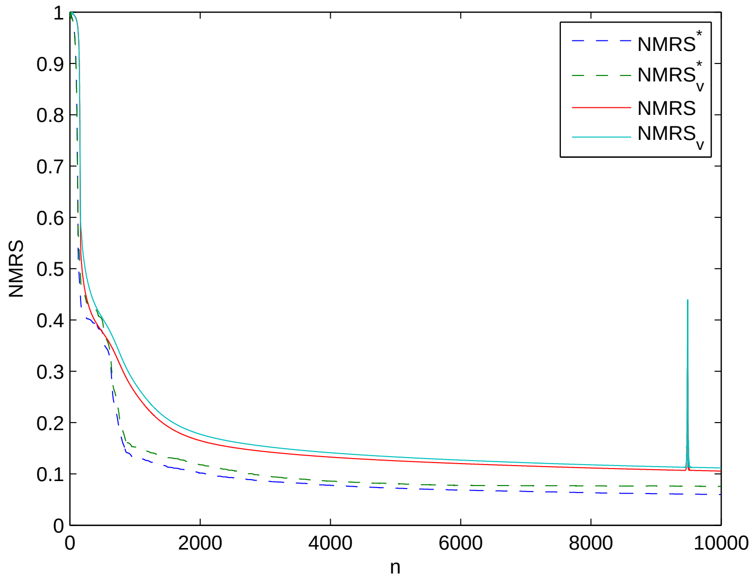


Slika 5.1: *Grafički prikaz rezultata učenja mreže na problemu identifikacije linearnog dinamičkog člana($NMRS^*$ i $NMRS_v^*$ su dobiveni korištenjem EBP algoritma sa adaptivnim koeficijentom učenja, dok su $NMRS$ i $NMRS_v$ dobiveni korištenjem EBP algoritma sa fiksnim koeficijentom učenja).*

Tablica 5.2: *Rezultati učenja mreže na problemu predviđanja ponašanja nelinearnog kaotičnog sustava opisanog Glass-Mackeyevom jednadžbom(EBP^* - EBP sa adaptivnim koeficijentom učenja;NMRS - NMRS učenja;$NMRS_v$ - NMRS validacije; $NMRS_{ti}$ - NMRS i-tog testa).*

	NMRS	$\mathbf{NMRS_v}$	korak	$\mathbf{NMRS_{t1}}$	$\mathbf{NMRS_{t2}}$	$\mathbf{NMRS_{t3}}$
EBP*	0.0599	0.0756	10000	0.0659	0.0671	0.0704
EBP	0.1056	0.1117	10000	0.1045	0.1050	0.1073

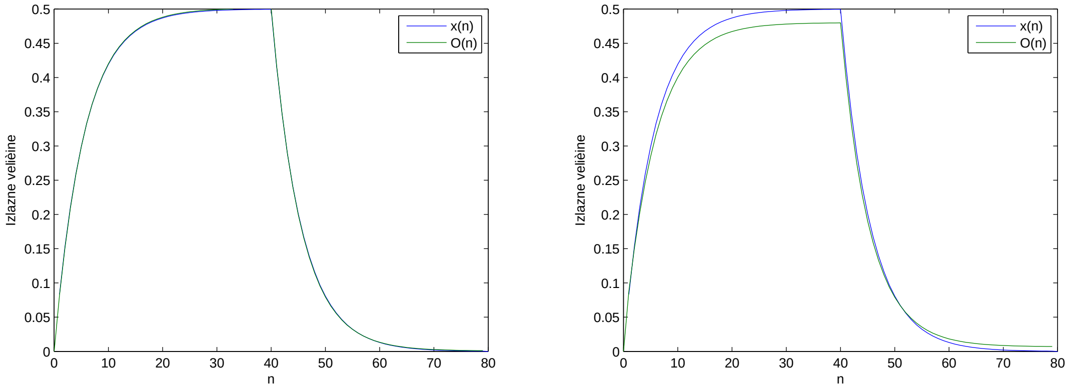
Na temelju dobivenih rezultata,kojima je prethodilo učenje različitim varijantama EBP algoritma iz jednakih početnih uvjeta možemo zaključiti da neuronske mreže na kojima se primjenjuje EBP algoritam sa adaptivnim koeficijentom učenja postižu veću točnost(manja vrijednost NMRS-a), te imaju bolju sposobnost generalizacije, za razliku od neuronskih mreža na kojima se primjenjuje standardni EBP algoritam učenja.



Slika 5.2: *Grafički prikaz rezultata učenja mreže na problemu predviđanja ponašanja nelinearnog kaotičnog sustava opisanog Glass-Mackeyevom jednadžbom($NMRS^*$ i $NMRS_v^*$ su dobiveni korištenjem EBP algoritma sa adaptivnim koeficijentom učenja, dok su $NMRS$ i $NMRS_v$ dobiveni korištenjem EBP algoritma sa fiksnim koeficijentom učenja)*

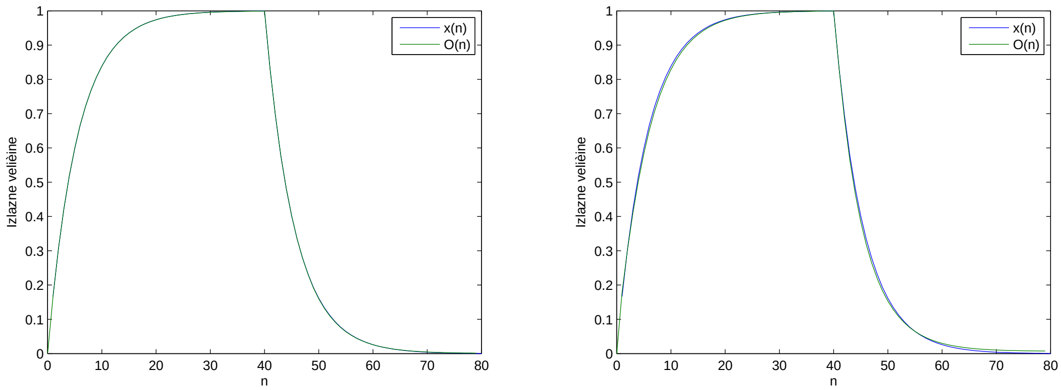
5.1 Testovi(P1 član)

Za izvođenje sljedećih testova uzeti su parametri mreže za koje se ostvaruje najmanji iznos $NMRS$-a validacije u zadanih 5000 koraka. Za EBP algoritam sa adaptivnim koeficijentom učenja(EBP*) najmanji iznos $NMRS$-a validacije je postignut zadnjem koraku, dok za EBP algoritam s fiksnim koeficijentom učenja je to postignuto u 3085-om koraku. Točnost za skok funkciju amplitude 0.5 je najlošija, zbog toga što je to pobuda koja je potpuno izvan učenog intervala.



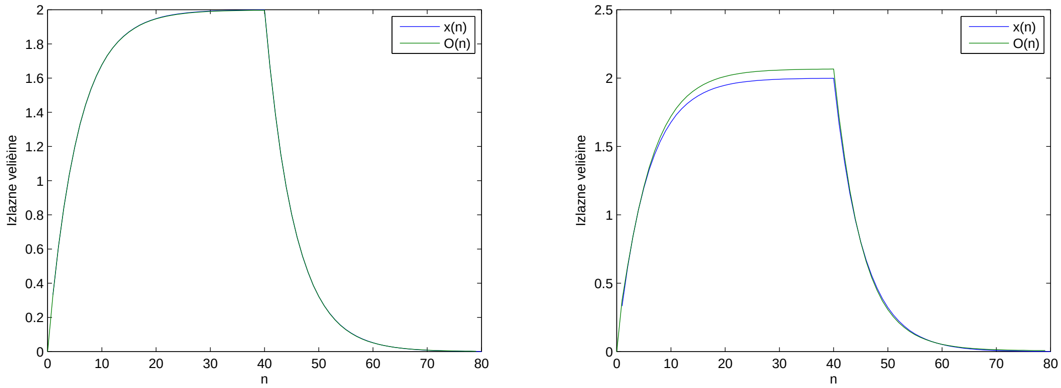
(a) EBP* (b) EBP

Slika 5.3: *Odzivi mreže na test 1*



(a) EBP* (b) EBP

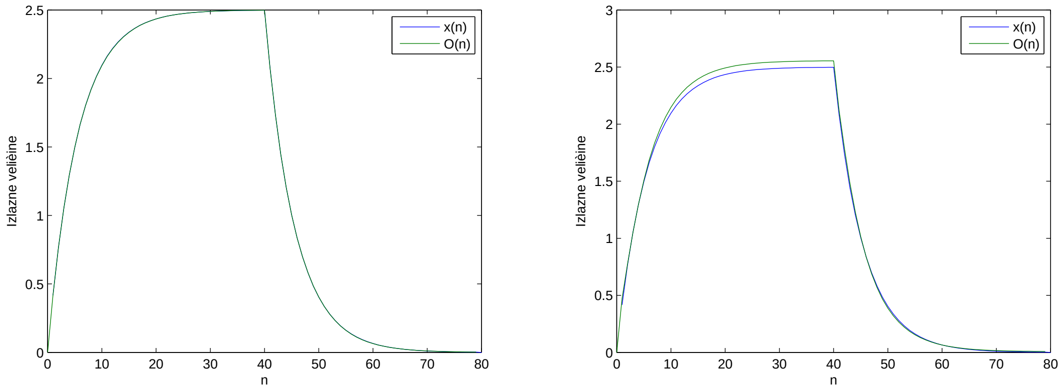
Slika 5.4: *Odzivi mreže na test 2*



(a) EBP*

(b) EBP

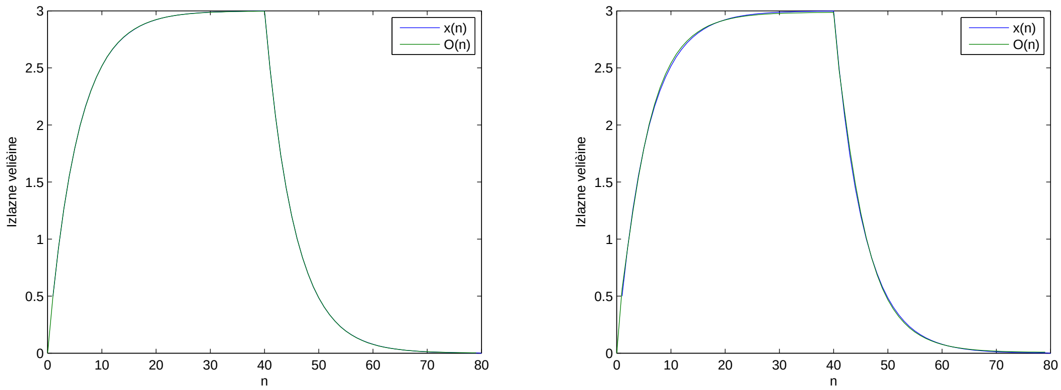
Slika 5.5: *Odzivi mreže na test 3*



(a) EBP*

(b) EBP

Slika 5.6: *Odzivi mreže na test 4*



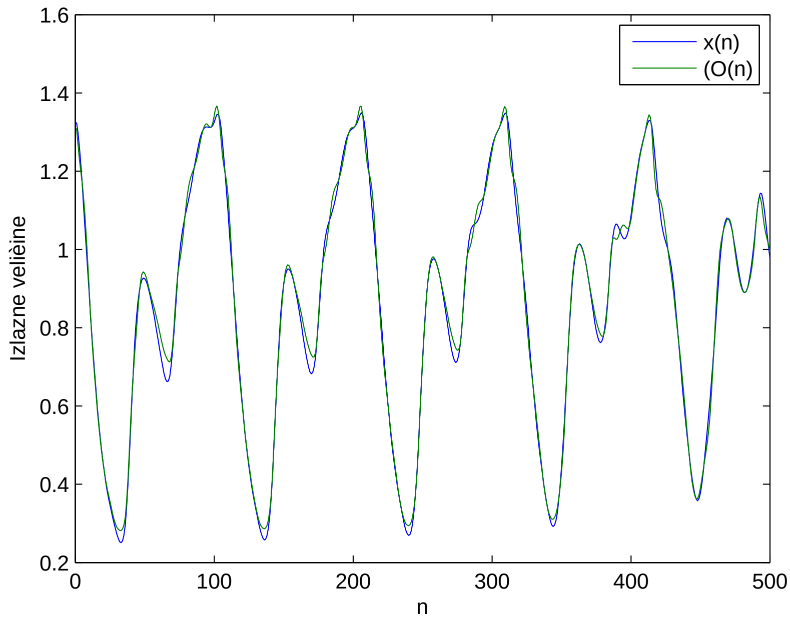
(a) EBP*

(b) EBP

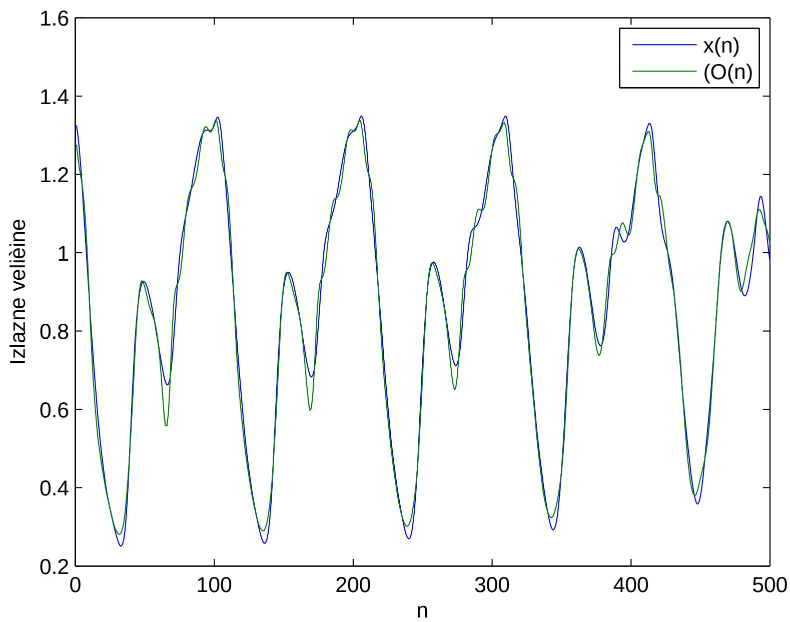
Slika 5.7: *Odzivi mreže na test 5*

5.2 Testovi(Glass-Mackey)

Za izvođenje sljedećih testova uzeti su parametri mreže za koje se ostvaruje najmanji iznos $NMRS$-a validacije u zadanih 10000 koraka, što se za oba dvije varijante EBP algoritma postiže u posljednjem koraku.

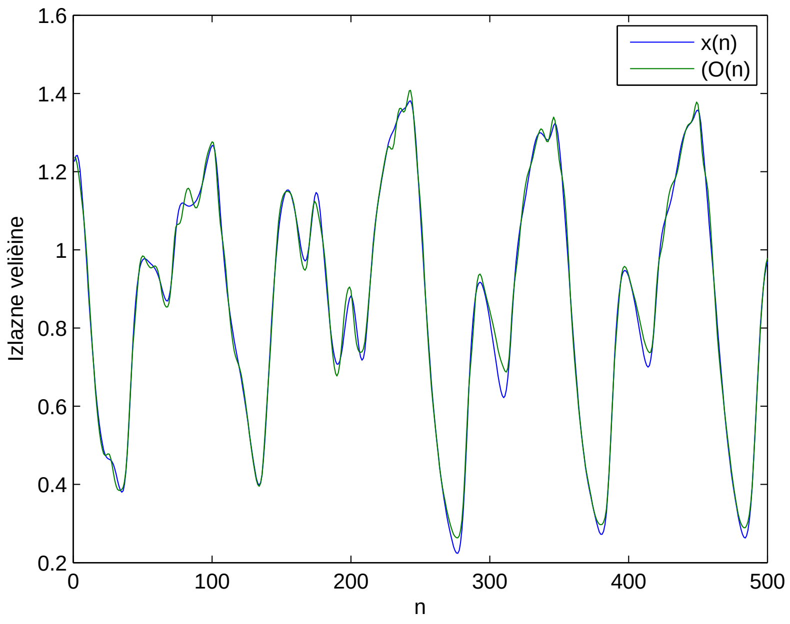


(a) EBP*

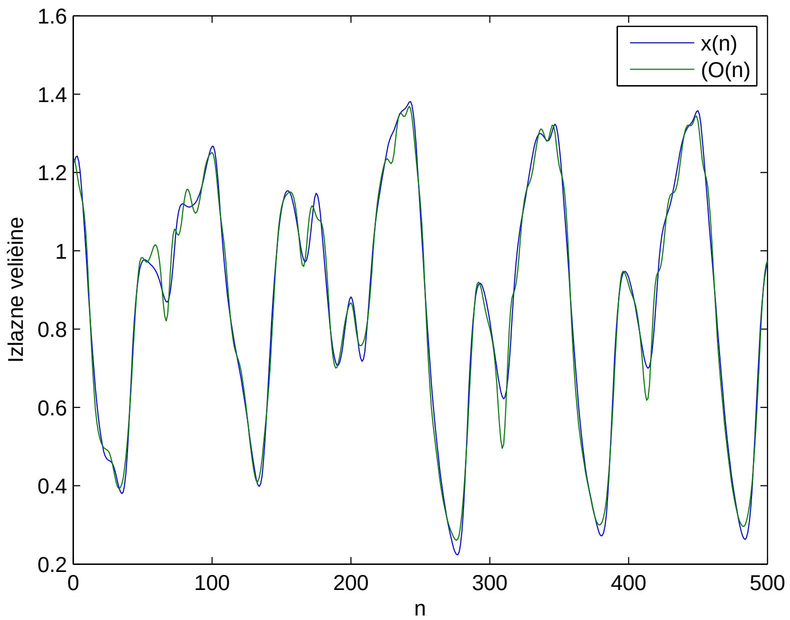


(b) EBP

Slika 5.8: *Odzivi mreže na test 1*

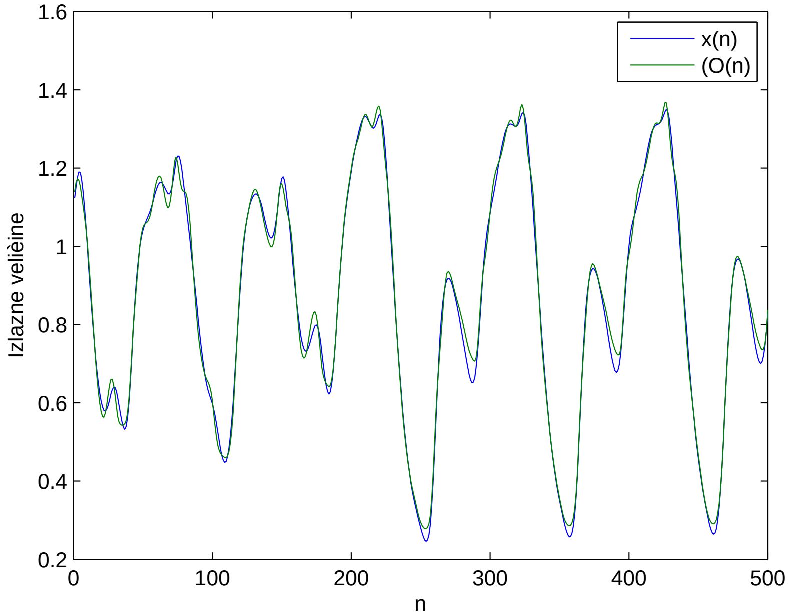


(a) EBP*

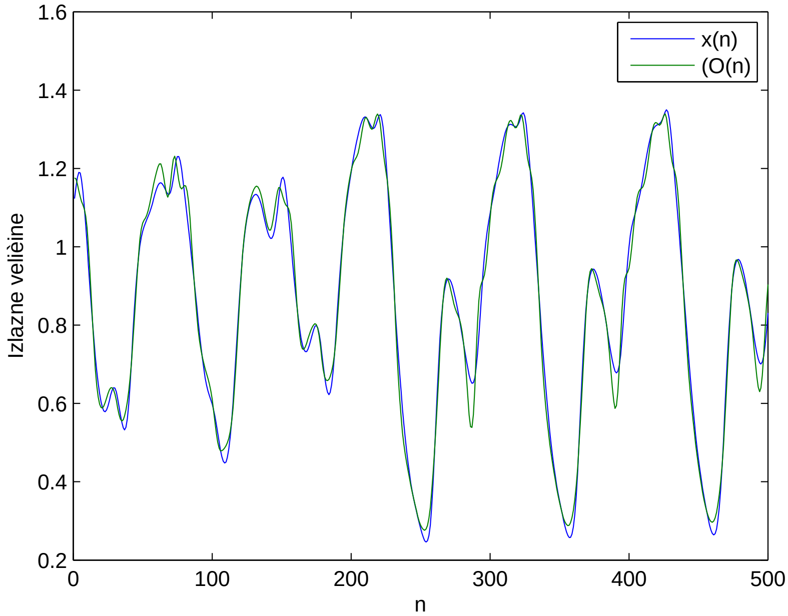


(b) EBP

Slika 5.9: *Odzivi mreže na test 2*



(a) EBP*



(b) EBP

Slika 5.10: *Odzivi mreže na test 3*

Poglavlje 6

Zaključak

Cilj ovoga rada bila je usporedba EBP algoritma sa fiksnim i adaptivnim koeficijentom učenja kroz niz standardnih testova, pri čemu su korištene troslojne perceptronske neuronske mreže sa unipolarnom sigmoidalnom aktivacijskom funkcijom. Strukture mreža su ovisile o problemu, te o uspješnosti učenja. Algoritmi učenja analizirani su primjenom dvije vrste testova. Prva vrsta testova definirana je jednadžbom dinamike P1 člana i različitim ulaznim funkcijama istog, dok je druga vrsta testova definirana Glass-Mackey jednadžbom i različitim početnim uvjetima iste. Za ocjenu uspješnosti algoritama učenja korišten je normalizirani korijen srednje pogreške. Na osnovi eksperimentalno dobivenih rezultata dolazimo do zaključka da učenje neuronske mreže primjenom EBP algoritma s adaptivnim koeficijentom učenja daje bolje rezultate u smislu točnosti i generalizacije od standardnog EBP algoritma.

Literatura

[1] D. Majetić, B. Novaković, M. Široki, *Umjetne neuronske mreže*, Fakultet strojarstva i brodogradnje Sveučilišta u Zagrebu, 2011.

[2] T. Baček, *Diplomski rad*, Fakultet strojarstva i brodogradnje Sveučilišta u Zagrebu, 2011.

[3] D. Majetić, *Neuronske mreže - podloge za predavanja*, Fakultet strojarstva i brodogradnje Sveučilišta u Zagrebu, 2011.

[4] http://java.zemris.fer.hr/nastava/nenr/knjiga-0.1.2012-11-26.pdf